ipd4000stdtxtTES-10

Defense Information Infrastructure (DII)

Common Operating Environment (COE)

Software Test Description (STD) and Software Test Report for the METOC Textual Observation API and Database Segments (MATXT and MDTXT)

Preliminary Release

Document Version 4.0

4 May 1998

Prepared for:
Naval Research Laboratory
Marine Meteorology Division
Monterey, CA

Prepared by:
Integrated Performance Decisions, Inc.
Middletown, RI

(This page intentionally left blank.)

Table of Contents

List of Tables

List of Figures

1 SCOPE

1.1 Identification

This Software Description Document (STD) and Software Test Report (STR) describes the test procedures and the report of the results used to verify the Textual Observation API Segment (MATXT), and the Textual Observation Database Segment (MDTXT), Versions 4.1 series, of the Tactical Environmental Support System (Next Century) [TESS(NC)] Meteorological and Oceanographic (METOC) Database. The MATXT and MDTXT segment provides APIs and a database for the storage, retrieval, and manipulation of textual METOC observations and bulletins. This software is designed to run under the Defense Information Infrastructure (DII) Common Operating Environment (COE), release 3.1, on a Hewlett-Packard computer running HP-UX 10.20 or a personal computer running the Microsoft Windows NT 4.0 operating system with Service Pack 3.

1.2 System Overview

The APIs described in this document form a portion of the METOC Database component of the TESS(NC) Program (NITES Version I). On 29 October 1996, the Oceanographer of the Navy issued a TESS Program Policy statement in letter 3140 Serial 961/6U570953, modifying the Program by calling for five seamless software versions which are Defense Information Infrastructure (DII) Common Operating Environment (COE) compliant, preferably, to level 5.

The five versions are:

- NITES Version I — The local data fusion center and principal METOC analysis and forecast system. (TESS(NC))
- NITES Version II — The subsystem on the JMCIS or GCCS systems (NITES/JMS)
- NITES Version III — The unclassified aviation forecast, briefing and display subsystem tailored to Naval METOC shore activities (currently satisfied by the Meteorological Integrated Data Display System (MIDDS))
- NITES Version IV — The Portable subsystem composed of independent PCs/workstations and modules for forecaster, satellite, communications, and IC4ISR functions (currently the Interim Mobile Oceanographic Support System (IMOSS))
- NITES Version V — Foreign Military Sales (currently satisfied by the Allied Environmental Support System (AESS))

NITES I acquires and assimilates various METOC data for use by US Navy and Marine Corps weather forecasters and tactical planners. NITES I provides these users METOC data, products, and applications necessary to support the warfighter in tactical operations and decision making. NITES I provides METOC data and products to NITES I and NITES II applications, as well as non-TESS(NC) systems requiring METOC data, in a heterogeneous networked computing environment.

The TESS(NC) Concept of Operations and system architecture require that the METOC Database be distributed both in terms of application access to METOC data and products and in terms of physical location of the data repositories. The organizational structure of the database is influenced by these requirements and the components of this distributed database are described below.

In accordance with DII COE database concepts, the METOC Database is composed of six DII COE compliant *shared database* segments. Associated with each shared database segment is an API segment. The segments are arranged by data type as follows:

Data Type	**Data Segment**	**API Segment**
Grid Fields	MDGRID	MAGRID
Latitude-longitude-time (LLT) Observations	MDLLT	MALLT
Textual Observations and Bulletins	MDTXT	MATXT
Remotely Sensed Data	MDREM	MAREM
Imagery and Product Data	MDIMG	MAIMG
Climatology Data	MDCLIM	MACLIM

A typical client-server installation is depicted in Figure 1 on the next page. This shows the shared database segments residing on a DII COE SHADE database server, with a NITES I or II client machine hosting the API segments. Communication between API segments and shared database segments is accomplished over the network using ANSI-standard Structured Query Language (SQL).

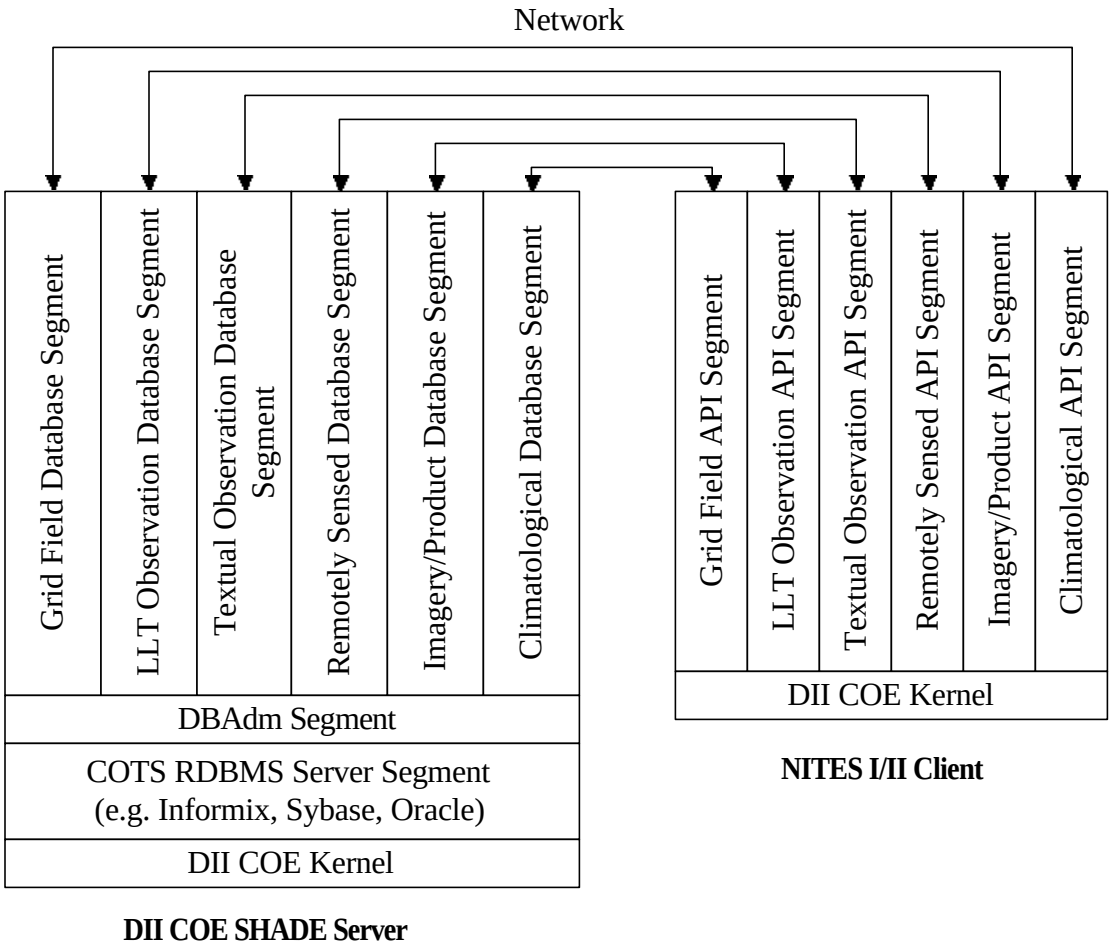

Figure 1. TESS(NC) METOC Database Conceptual Organization

The MATXT and MDTXT segments deal with textual observations and bulletins. Textual observational data are primarily ASCII formatted forecasts or bulletin/warning oriented messages. Textual observation data can be associated with a specific geographic point and time, but more generally are associated with a geographical area or region. Types include Forecast Reports, Warnings, and Notices. Depending on the type of textual observation, the reporting station or organization and the area or region affected is decoded and stored along with the textual portion of the message. Textual observation data are typically displayed as text by a client application.

(This page intentionally left blank.)

2 REFERENCED DOCUMENTS

2.1 Government Documents

STANDARDS

MIL-STD-498 *Software Development and Documentation*
5 December 1994

SPECIFICATIONS

Unnumbered *Performance Specification (PS) for the Tactical Environmental*
5 December 1997 *Support System/Next Century TESS(NC) (AN/UMK-3)*

Unnumbered *Software Requirements Specification for the Tactical Environmental*
30 September 1997 *Support System/Next Century [TESS(3)/NC] Meteorological and Oceanographic (METOC) Database*, Space and Naval Warfare Systems Command, Environmental Systems Program Office (SPAWAR PMW-185), Washington, DC

OTHER DOCUMENTS

Unnumbered *Database Design Description for the Tactical Environmental Support*
30 September 1997 *System/Next Century [TESS(3)/NC)] Meteorological and Oceanographic (METOC) Database*, Space and Naval Warfare Systems Command, Environmental Systems Program Office (SPAWAR PMW-185), Washington, DC

DII.COE.DocReqs-5 *Defense Information Infrastructure (DII) Common Operating*
29 April 1997 *Environment (COE) Developer Documentation Requirements, Version 1.0*

Department of the Air Force, Headquarters Air Weather Service, Scott AFB, ILL

AWSR 105-2 *Weather Communications Policies and Procedures*
24 August 1990

Naval Research Laboratory, Marine Meteorology Division, Monterey, CA

Unnumbered 30 March 1998	*Programming Manual (PM) for the Textual Observation API Segment (MATXT) of the Tactical Environmental Support System (Next Century) [TESS(NC)] Meteorological and Oceanographic (METOC) Database*
Unnumbered 30 March 1998	*Application Program Interface Reference Manual (APIRM) for the Textual Observation API Segment (MATXT) of the Tactical Environmental Support System (Next Century) [TESS(NC)] Meteorological and Oceanographic (METOC) Database*
Unnumbered 30 March 1998	*Installation Procedures (IP) for the Textual Observation API Segment (MATXT) of the Tactical Environmental Support System (Next Century) [TESS(NC)] Meteorological and Oceanographic (METOC) Database*
Unnumbered 30 March 1998	*Installation Procedures (IP) for the Textual Observation Database Segment (MDTXT) of the Tactical Environmental Support System (Next Century) [TESS(NC)] Meteorological and Oceanographic (METOC) Database*
Unnumbered 30 March 1998	*Software Version Description (SVD) for the Textual Observation API Segment (MATXT) of the Tactical Environmental Support System (Next Century) [TESS(NC)] Meteorological and Oceanographic (METOC) Database*
Unnumbered 30 March 1998	*Software Version Description (SVD) for the Textual Observation Database Segment (MDTXT) of the Tactical Environmental Support System (Next Century) [TESS(NC)] Meteorological and Oceanographic (METOC) Database*

2.2 Non-Government Documents

World Meteorological Organization, Geneva, Switzerland

WMO-306 1995	*Manual On Codes*

3 TEST PREPARATION

3.1 MATXT and MDTXT Segments Test Preparation

Segment testing for each of the tests and test cases identified in Section 4 is conducted in the TESS(NC) target hardware and software environment. The approach for all testing with the exception of the installation tests (Section 4.1) is performed using software test driver programs with file or tester provided input data. The goal is to automate the testing to the greatest extent possible (within the schedule constraints) to facilitate general, integration, and regression testing of the segments. The test driver programs used are the same as the test driver programs delivered as part of the API segment delivery. Following API segment installation, the programs are located in the subdirectory: **h/MATXT/Integ/TestSuite**.

3.1.1 Hardware Preparation

To run the tests described herein, the tester must have installed the database segment (MDTXT) on the test server machine where the DII COE RDBMS and DB administration tools reside. Depending on the test client machine and test case, the tester must also have installed the API segments (MATXT), on the test server machine, the test client machine, or in some cases, both machines. The hardware requirements for the test are described in Section 3.1.1 of the *Installation Procedure (IP) for the Textual Observation Database Segment (MDTXT)* and the *Installation Procedure (IP) for the Textual Observation API Segment (MDTXT)*. Testing of the APIs in the networked client/server configuration (i.e. PC/Windows NT client, HP-UX database server) requires that both client and server machines are connected on an IEEE 802.3/5 local area network supporting the TCP/IP protocol.

3.1.2 Software Preparation

With the exception of the segment installation tests, the textual observation segment tests are performed using semi-automated software test driver programs. These test driver programs are HP-UX and Windows NT portable, command line invokable, executable. The programs rely on ASCII text file and manually entered command line inputs for test case data and control. The program output can be written directly to screen or redirected to files for review or permanent capture. Test drivers have been written for both static and dynamic API library verification. Each test driver supports multiple test cases through single or batch processing of the text files. Manual testing is also supported using the command line parameters.

3.1.2.1 Using the Test Driver Programs

The following table identifies the executable test driver programs used to support textual observation testing.

Table 3-1. Textual Observation Test Drivers

Test Driver Name[1]	Test Driver Description	Test
MATXTtestIngest_d/_l	Ingest (stores) the input textual observations to the database.	4.2
MATXTtestGetByID_d/_l	Retrieves (selects) textual observation from the database according to a specific record ID. Textual observations must be stored (ingested) prior to running this test.	4.3
MATXTtestGetByQuery_d/_l	Retrieves any number of textual observations from the database for a specified criteria. Textual observations must be stored (ingested) prior to running this test.	4.4
MATXTtestCatalog_d/_l	Retrieves a catalog listing (summary list) of textual observations from the database for a specified criteria. Textual observations must be stored (ingested) prior to running this test.	4.5
MATXTtestUpdateByID_d/_l	Updates a single textual observation in the database with specific data as input by the tester. Textual observations must be stored (ingested) prior to running this test.	4.6
MATXTtestDelete_d/_l	Deletes the textual observations from the database for a record ID. Textual observations must be stored (ingested) prior to running this test.	4.7

Note 1: Driver program names end with either a _d or _l, where _d = dynamic API library, and _l = static API library. Programs are otherwise functionally identical. Prior to running the dynamic API test drivers, an environmental variable must be set after a new installation of the MATXT segment. This procedure is covered in Section 3.1.3 of this document.

3.1.2.2 Command Line Arguments

The command line arguments permit the tester to control the test program, test program inputs, test program output, and exercise the test cases developed for the segment. These arguments also allows the tester to manual enter test data and conditions to exercise segment functionality. Each of the driver programs supports the same set of command line parameter inputs. The following

table describes the arguments. The *italicized* text denotes the sample name of a file given by the user and any naming convention is up to the user.

Table 3-2. Textual Observation Command Line Arguments

Argument	Description
-h	Displays help information about the specific test driver program invoked. e.g., MATXTtestIngest **-h**
-d	Turns on the debug macros (DPRINTS) within the APIs causing the display of debug information at runtime. e.g., MATXTtestIngest –b TESTDATA/INGEST **-d**
-l <file name>	Saves the manually entered inputs to a file, which can be used to run automated testing in the batch (-b) processing mode. e.g., MATXTtestIngest **–l** *ingtest2*
-b <directory path>	Turns on test driver batch processing mode. Batch processing mode causes the test driver to run the test cases associated with the input files located in the specified directory path. One or more input files can be located in the directory path. e.g., MATXTtestIngest **–b** TESTDATA/INGEST
-f <file name>	Used in conjunction with the MATXTtestIngest driver to assign WMO formatted textual observation to the ingested observation. The STRINGFILE subdirectory contains the ASCII text data that represents the text observation. E.g., MATXTtestIngest –l Ingest1 **–f** /STRINGFILE/text _info

3.1.2.3 Test Input Data

Test driver program inputs are provided either manually through tester interaction with the test program or through the test case oriented ASCII text input files. The input files are delivered with the API segment and loaded onto the target system when the segment is installed. The files are located under the **/h/MATXT/Integ/TestSuite/TESTDATA** path of the target system. The following table identifies the subdirectory name and applicable test under the TESTDATA path.

Table 3-3. Test Driver Subdirectories and Names

Test	Subdirectory	Description
4.2	INGEST/	This subdirectory contains the textual observation ingest test case data files. These files are used in conjunction with the MATXTtestIngest test drivers.
4.3	GETBYID/	This subdirectory contains the textual observation test case files with the parameters required to retrieve observations using the database record ID. These files are used in conjunction with the MATXTtestGetByID test drivers.
4.4	GET/	This subdirectory contains the textual observation test case files with the parameters required to retrieve one or more textual observations for a specified criteria. These files are used in conjunction with the MATXTtestGetByQuery test drivers.
4.5	CAT/	This subdirectory contains the textual observation test case files with the parameters required to retrieve a catalog listing of one or more textual observations for a specified criteria. These files are used in conjunction with the MATXTtestCatalog test drivers.
4.6	UPDATE/	This subdirectory contains the textual observation test case files with the data fields used to update existing observations stored in the database. These files are used in conjunction with the MATXTtestUpdate test drivers.
4.7	DELETE/	This subdirectory contains the textual observation test case files providing the deletion criteria used to delete observations from the database. in conjunction with the These files are used in conjunction with the MATXTtestDeleteByID test drivers.
N/A	STRING FILE/	This subdirectory stores ASCII text files that contain the WMO formatted textual observation message text. These files are used in conjunction with the MATXTtestIngest test drivers to populate the database record with message text.

3.1.2.4 Test Output Data

To facilitate review of the desired test cases. Once a test drive is executed on an xterm or console window (see Section 3.1.3 for invoking a window on the HP-UX or NT systems), the user will be able to review debug deprints and/or the status of the driver results. If desired, the test team can

redirect these out puts to a file of a specific name for later review. For example, running the ingest test driver with batch (-b) processing, with the INGEST subdirectory test case, with debug deprints, and redirected to the file named *test1*. At the command line the following would be entered at the prompt (/h/MATXT/Integ/TestSuite>) and executed by selecting the <Enter> button:

- MATXTtestIngest_l –b TESTDATA/INGEST -d >test1

All test results which supports this document are furnished in the on a 3.5" floppy disk in a "text" format (the debug deprint are not provided). Test Results are discussed in **Appendix B** of this document.

In addition, to ensure the test case ingested the data in the databases on the HP-UX machine. Go to the target machine, open an xterm window and follow the steps in Section 3.1.3 to use DB Access, which will verify that the textual observations were ingested. Each observation can be reviewed to view the various data entries in each.

3.1.3 Other Preparations

3.1.3.1 Establishing an Xterm or console window on the HP-UX and NT platforms

In order to efficiently test the database and API segments, semi-automated software test driver programs were developed. These programs are command line invokable executables that use manual or file inputs to generate redirectable outputs to the display. To operate these driver programs in the DII COE 3.1 software environment requires that an "Xterm" or console window be made available. The following procedures describe how to create the console windows for the HP-UX and Windows NT test environments.

In some of the test cases it maybe necessary for the user to verify the data has been ingested, updated, or deleted in the database located on the HP-UX system. This is accomplished using an xterm window and DB Access with the steps discussed in the Section below.

3.1.3.2 Establishing a console window for a DII COE 3.1 HP-UX System

Log in as *sysadmin* and perform the following steps:

1. Click on the **Application Manager** icon on toolbar.
2. Double-click on the **Desktop_Apps** icon.
3. Double-click on the **Create Action** icon.
4. Enter `xterm` in the **Action Name** field.
5. Click **Find Set...** in the Action Icons panel.

6. In **Icon Folders** list, double-click on **/usr/dt/appconfig/icons/C.**
7. Scroll **Icon Files** list down to the **Dtxterm** icon (a terminal with an X).
8. Click on the **Dtxterm** icon.
9. Click on the **Ok** button.
10. Enter **/usr/bin/X11/xterm -sb -sl 800** in Command field.
11. Enter **This is an xterm** in Help Text field.
12. Select **File/Save** from window menu bar.
13. You should see a **Create Action - Confirmation** window appear.
14. Click **Ok.**
15. Close **Create Action** window.
16. Close **Application Manager** window.
17. Click on the **Home Folder** icon on toolbar.
18. You should see the new action in your folder, and double-click on the new action to launch the folder.

If desired this icon can be installed into the "Personal Applications" pop-up menu panel on the toolbar. This is accomplished by:

1. Click on the **Home Folder** icon on toolbar.
2. Click on the **Personal Application** panel "up arrow" button (above the icon) and drag your new action and drop it on the **Install Icon** button.
3. You should see your new icon appear in the panel.
4. You can move the new icon so it is always visible on the toolbar by right-clicking on the new icon and select **Copy to Main Panel.**

3.1.3.3 Establishing a console window for a Windows NT 4.0 System

The following steps are required to initialize an MS DOS console window on the Windows NT system.

Login as the appropriate user (site dependent) and perform the following steps:

1. Click on the **Start** button at the lower left hand portion of the window.
2. Select **Programs** directly followed by **MS-DOS Prompt**.
3. A *MS-DOS PROMPT* window will be displayed with the DOS command line (C:\) prompt.

3.1.3.4 Setting the Environmental Variable to Run the Dynamic Library API

Before testing the dynamic (shared) library API (_d), the tester must ensure that the environmental variable is set after each fresh installation of the MATXT segment. This is accomplished my opening an xterm and console window on the HP-UX and NT machines, respectively.

The following steps are required to complete this process:

HP-UX:

1. Open an xterm window.
2. Set the path on the HP-UX by typing:
   `setenv SHLIB_PATH ${SHLIB_PATH}:/h/MATXT/bin` <Enter>

NT:

1. Open a DOS window.
2. Set the path by typing:
   `set PATH=%PATH%;c:/h/MATXT/bin` <Enter>

3.1.3.5 Reviewing the Content of the Database using DB Access

When running the test drivers and associated test cases for Ingest, Update, Delete it may be necessary to review the affected changes in the database located on the HP-UX machine. This is facilitated by running the Informix DB Access tool by opening an xterm window on the target machine and completing the following steps:

1. In the xterm, change directories to the informix bin directory:
   `>cd /opt/informix/bin`
2. Set the environmental variable in the xterm by typing:
   `>setenv TERM vt100`
3. Run dbaccess by typing:
   `>dbaccess`
4. Once in dbaccess, select **Query Language** from menu (default selection) and press < return>
5. Use up/down arrows to select database of interest (e.g., mdtxt) then press <return>
6. The user is provided several menu options. These options can be selected by either typing the first letter of the option (e.g., typing `I` for Info), or using the arrow keys and <return> to select an option.
7. To view the list of data sets in the database by name, select **Info** and the list of data sets will be displayed. Select <return> then **E** for exit.

8. To view records in a data set (the data set name is required), select **New** and type:
   `select * from datasetname` and press <escape> and then select **Run**.
9. This result will display the ingested records stored in the informix database as applied by the ingest test case. The list will show the exact data fields for each observation stored in the database. If there is more than one page, select **Next** from the menu until all records have been displayed.
10. To exit dbaccess, use the **Exit** menu selection. User may need to exit several menu levels before actually exiting dbaccess.

(This page intentionally left blank.)

4 TEST DESCRIPTIONS

4.1 MDTXT and MATXT Segment Installation Test

The following test cases comprise a segment installation test to verify that the textual observation database and API segments install correctly in the target hardware and software environment.

4.1.1 MDTXT Database Segment Installation Test Case

This test case verifies the correct installation of the MDTXT database segment. MDTXT will be installed using the DII COE provided installation tools on the HP-UX target platform.

4.1.1.1 Prerequisite Conditions

The prerequisite conditions for this test case are defined in Section 3.0 of the *Installation Procedure (IP) for the Textual Observation Database Segment (MDTXT) of the Tactical Environmental Support system Next Century [TESS(NC) Meteorological and Oceanographic (METOC) Database*, version 4.1 or later (herein referred to as MDTXT IP).

4.1.1.2 Test Inputs

There are no test inputs for this test case other than the operator actions identified in Section 4.0 of the MDTXT IP.

4.1.1.3 Expected Test Results

The Segment Installer window will display **METOC Text Observation Database Segment** in the Currently Installed Segments Section of the window (See Section 4.0 of the MDTXT IP).

4.1.1.4 Criteria for Evaluating Results

The Segment Installer tool determines and indicates successful installation of the segment to the tester.

4.1.1.5 Test Procedure

The test procedure is identical to the segment installation instructions provided in Section 4.0 of the MDTXT IP.

4.1.1.6 Assumptions and Constraints

This test assumes the target hardware is operating correctly and configured with the operating and application software identified in Sections 3.1 and 3.2 in the MDTXT IP.

4.1.2 MATXT API Segment HP Installation Test Case

This test case verifies the correct installation of the HP-UX MATXT API segment. MATXT will be installed using the DII COE provided installation tools on the HP-UX target platform. The MATXT API Segment for HP provides both the dynamic link and static link libraries when installed.

4.1.2.1 Prerequisite Conditions

The prerequisite conditions for this test case are defined in Section 3.0 of the *Installation Procedure (IP) for the Textual Observation API Segment (MATXT) of the Tactical Environmental Support system Next Century [TESS(NC) Meteorological and Oceanographic (METOC) Database*, version 4.1 or later (herein referred to as MATXT IP).

4.1.2.2 Test Inputs

There are no test inputs for this test case other than the operator actions identified in Section 4.0 of the MATXT IP.

4.1.2.3 Expected Test Results

The Segment Installer window will display **METOC Text Observation API Segment** in the Currently Installed Segments Section of the window (See Section 4.0 of the MATXT IP).

4.1.2.4 Criteria for Evaluating Results

The Segment Installer tool determines and indicates successful installation of the segment to the tester.

4.1.2.5 Test Procedure

The test procedure is identical to the segment installation instructions provided in Section 4.0 of the MATXT IP.

4.1.2.6 Assumptions and Constraints

This test assumes the target hardware is operating correctly and configured with the operating and application software identified in Sections 3.1 and 3.2 in the MATXT IP.

4.1.3 MATXT API Segment Windows NT Installation Test Case

This test case verifies the correct installation of the Windows NT version of the MATXT API segment. MATXT will be installed using the InstallShield™ software provided with the MATXT Windows NT Segment. The MATXT API Segment for Windows NT provides both the dynamic link and static link libraries when installed.

4.1.3.1 Prerequisite Conditions

The prerequisite conditions for this test case are defined in Section 3.0 of the *Installation Procedure (IP) for the Textual Observation API Segment (MATXT) of the Tactical Environmental Support system Next Century [TESS(NC) Meteorological and Oceanographic (METOC) Database*, version 4.1 or later (herein referred to as MATXT IP).

4.1.3.2 Test Inputs

There are no test inputs for this test case other than the operator actions identified in Section 4.0 of the MATXT IP.

4.1.3.3 Expected Test Results

The InstallShield™ installation program will display the Installation Complete dialogue box. A directory listing of the C:\h\MATXT directory and subdirectories will display the dynamic link libraries, static link libraries, API test drivers, and test datasets installed with the segment. Note: The Windows NT Explorer application can be used to view the contents of the directories. Section 4.4 of the MATXT IP lists the installation directories and contents.

4.1.3.4 Criteria for Evaluating Results

The InstallShield™ installation program determines and indicates successful installation of the segment to the tester. Additionally, the contents of the **C:\h\MATXT** directory and subdirectories should match the list referenced in Section 4.4 of the IP document.

4.1.3.5 Test Procedure

The test procedure is identical to the segment installation instructions provided in Section 4.0 of the MATXT IP.

4.1.3.6 Assumptions and Constraints

This test assumes the target hardware is operating correctly and configured with the operating and application software identified in Sections 3.1 and 3.2 in the MATXT IP.

4.2 Textual Observation Data Ingest Test

The following test cases verify that the MDTXT database and MATXT API segments support the ingest and store textual observations.

4.2.1 Ingest Textual Observation Test Case

This test case will verify that MDTXT and MATXT correctly store the Textual Observation messages identified in Table 3.2-7 of the METOC Database SRS. The test will demonstrate the storing of each message type/subtype defined by the Table A and B1 designators and the storing of associated descriptive information.

In addition the storage of the these textual observation types will have support the testing of erroneous observations; geographic parameters; and Year 2000 (Y2K) issues.

4.2.1.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 if this document.

3. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

4. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

4.2.1.2 Test Inputs

The test inputs necessary for this test case are provided in the **INGEST** test case driver files described in **Appendix A** of this document. These files provide the descriptive parameters and textual observation data required for ingest of textual observations to the database. The textual observation input data matches the formats defined in WMO-386 for the textual observations supported by MDTXT. These files vary by type (Table A Designator) and subtype (Table B1 Designator) combination MDTXT SRS Table 3.2-7.

4.2.1.3 Expected Test Results

The MDTXT database will contain correctly formatted textual observation messages and associated descriptive data corresponding to the input textual observation messages. Unique record IDs will be generated for each message type/subtype record stored in the database. Each textual observation record will contain the entire contents of the original message. In addition specific observations will not be installed to test how observations with erroneous data is handled. The detailed expected test results are provided in **Appendix B** of this document.

4.2.1.4 Criteria for Evaluating Results

The test case outputs derived from the test driver and tester use of the database access tool (e.g., Informix DB Access discussed in Section 3.1.3) must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The database is directly tested using the database access (dbaccess) tool to query and verify observation data entries have been stored properly and completely. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.2.1.5 Test Procedure

Once the Prerequisite Conditions (4.2.1.1) for the test have been met, the **MATXTtestIngest** test driver program is run by the tester. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.2.1.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.2.2 Multi User Ingest Textual Observation Test Case

This test case will verify that MDTXT and MATXT correctly store the Textual Observation messages identified in Table 3.2-7 of the METOC Database SRS while simultaneously executing the operation at booth the HP-UX and Windows NT machines without error. The test will demonstrate the storing of each message type/subtype defined by the Table A and B1 designators and the storing of associated descriptive information.

4.2.2.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document.

3. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

4. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "Up" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

4.2.2.2 Test Inputs

The test inputs necessary for this test case are provided in the **INGEST** test case driver files described in **Appendix A** of this document. These files provide the descriptive parameters and textual observation data required for ingest of textual observations to the database. The textual observation input data matches the formats defined in WMO-386 for the textual observations supported by MDTXT. These files vary by type (Table A Designator) and subtype (Table B1 Designator) combination MDTXT SRS Table 3.2-7.

4.2.2.3 Expected Test Results

The MDTXT database will contain correctly formatted textual observation messages and associated descriptive data corresponding to the input textual observation messages. Since the HP-UX and Windows NT machines are populating the targeted database, twice as many textual observations will be ingested compare to a single ingest discussed in Section 4.2.1. Unique record IDs will be generated for each message type/subtype record stored in the database. Each textual observation record will contain the entire contents of the original message. The detailed expected test results are provided in **Appendix B** of this document.

4.2.2.4 Criteria for Evaluating Results

The test case outputs derived from the test driver and tester use of the database access tool (e.g., Informix DB Access discussed in Section 3.1.3) must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The database is directly tested using the database access (dbaccess) tool to query and verify observation data entries have been stored properly and completely. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.2.2.5 Test Procedure

Once the Prerequisite Conditions (4.2.2.1) for the test have been met, the **MATXTtestIngest** test driver program is run by the tester on both the HP-UX and Windows NT platforms simultaneously. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.2.2.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.3 Textual Observation Get By ID Test

The following test cases verify that the MDTXT database and MATXT API segments supporting the retrieval of a single observation from the database.

4.3.1 Retrieve Textual Observation with the Get By ID Test Case

This test case will verify that MDTXT and MATXT correctly retrieves a single textual observation messages identified in Table 3.2-7 of the METOC Database SRS. The test will demonstrate the retrieving of a single observation using a series of test cases.

4.3.1.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document.

3. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

4. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

5. The database must have textual observation data available. It may be desirable to run a simulated or real data ingest prior to running this test case.

4.3.1.2 Test Inputs

The test inputs necessary for this test case are provided in the **GETBYID** test case driver files described in **Appendix A** of this document. These files provide the Record ID required for retrieval of a single textual observations from the database.

4.3.1.3 Expected Test Results

The API will retrieve correctly formatted textual observation messages (one per case) and associated descriptive data corresponding to the input textual observation messages. The detailed expected test results are provided in **Appendix B** of this document.

4.3.1.4 Criteria for Evaluating Results

The test case outputs derived from the test driver must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.3.1.5 Test Procedure

Once the Prerequisite Conditions (4.3.1.1) for the test have been met, the **MATXTtestGetByID** test driver program is run by the tester. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.3.1.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.3.2 Retrieve Textual Observation with Get By ID Error Test Case

This test case will verify that MDTXT and MATXT will not retrieves a single textual observation messages identified in Table 3.2-7 of the METOC Database SRS that has an erroneous Record ID.

The test will demonstrate that an erroneous Record ID will not retrieve a single observation using a series of test cases.

4.3.2.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document.

3. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

4. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

5. The database must have textual observation data available. It may be desirable to run a simulated or real data ingest prior to running this test case.

4.3.2.2 Test Inputs

The test inputs necessary for this test case are provided in the **GETBYID** test case driver files described in **Appendix A** of this document. These files provide the Record ID required for retrieval of a single textual observations from the database.

4.3.2.3 Expected Test Results

The API will not retrieve a textual observation messages (one per case) and associated descriptive data corresponding to the input textual observation messages. The detailed expected test results are provided in **Appendix B** of this document.

4.3.2.4 Criteria for Evaluating Results

The test case outputs derived from the test driver must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.3.2.5 Test Procedure

Once the Prerequisite Conditions (4.3.2.1) for the test have been met, the **MATXTtestGetByID** test driver program is run by the tester. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.3.2.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.3.3 Simultaneous Retrieve of a Textual Observation with the Get By ID Test Case

This test case will verify that MDTXT and MATXT correctly retrieves a single textual observation messages identified in Table 3.2-7 of the METOC Database SRS when executed from the HP-UX and Windows NT machine simultaneously. The test will demonstrate the retrieving of a single observation on both systems with out error using a series of identical test cases.

4.3.3.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows

are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.13. of this document.

3. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

4. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

5. The database must have textual observation data available. It may be desirable to run a simulated or real data ingest prior to running this test case.

4.3.3.2 Test Inputs

The test inputs necessary for this test case are provided in the **GETBYID** test case driver files described in **Appendix A** of this document. These files provide the Record ID required for retrieval of a single textual observations from the database.

4.3.3.3 Expected Test Results

The API will retrieve correctly formatted textual observation messages (one per case) and associated descriptive data corresponding to the input textual observation messages. The detailed expected test results are provided in **Appendix B** of this document.

4.3.3.4 Criteria for Evaluating Results

The test case outputs derived from the test driver must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.3.3.5 Test Procedure

Once the Prerequisite Conditions (4.3.3.1) for the test have been met, the **MATXTtestGetByID** test driver program is run by the tester on both the HP-UX and Windows NT machines simultaneously. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.3.3.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.4 Textual Observation Get By Query

The following test cases verify that the MDTXT database and MATXT API segments supporting the retrieval of multiple observations from the database.

4.4.1 Retrieve Textual Observations with the Get By Query Test Case

This test case will verify that MDTXT and MATXT correctly retrieves multiple textual observation messages identified in Table 3.2-7 of the METOC Database SRS. The test will demonstrate the retrieving of a multiple textual observation using a series of test cases.

4.4.1.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document.

3. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

4. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

5. The database must have textual observation data available. It may be desirable to run a simulated or real data ingest prior to running this test case.

4.4.1.2 Test Inputs

The test inputs necessary for this test case are provided in the **GET** test case driver files described in **Appendix A** of this document. These files provide data required to query each input field. These fields cover, Type, Report Times, Valid Times, Lat/Lon, Originating Site, and the Method of Receipt, which are used in combination to retrieve multiple textual observations from the database.

4.4.1.3 Expected Test Results

The API will retrieve correctly formatted textual observation messages and associated descriptive data corresponding to the input textual observation messages. The detailed expected test results are provided in **Appendix B** of this document.

4.4.1.4 Criteria for Evaluating Results

The test case outputs derived from the test driver must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.4.1.5 Test Procedure

Once the Prerequisite Conditions (4.4.1.1) for the test have been met, the MATXTtestGetByQuery test driver program is run by the tester. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.4.1.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.4.2 Retrieve Textual Observations with Get By Query Test Case, Geographic Area

This test case will verify that MDTXT and MATXT correctly retrieves multiple textual observation messages identified in Table 3.2-7 of the METOC Database SRS. The test will demonstrate the retrieving of a multiple textual observation using a series of test cases. These test cases are geared to exercise the retrieval of specific geographic areas, which may be confusing for the database especially when making requests across the equator, international date, line and the Greenwich Prime Meridian.

4.4.2.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document.

3. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

4. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

5. The database must have textual observation data available. It may be desirable to run a simulated or real data ingest prior to running this test case.

4.4.2.2 Test Inputs

The test inputs necessary for this test case are provided in the **GET** test case driver files described in **Appendix A** of this document. These fields cover, Type, Report Times, Valid Times, Lat/Lon, Originating Site, and the Method of Receipt, which are used in combination to retrieve multiple textual observations from the database.

4.4.2.3 Expected Test Results

The API will retrieve correctly formatted textual observation messages and associated descriptive data corresponding to the input textual observation messages. The detailed expected test results are provided in **Appendix B** of this document.

4.4.2.4 Criteria for Evaluating Results

The test case outputs derived from the test driver must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.4.2.5 Test Procedure

Once the Prerequisite Conditions (4.4.2.1) for the test have been met, the MATXTtestGetByQuery test driver program is run by the tester. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.4.2.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.4.3 Retrieve Textual Observations with Get Buy Query Test Case, Year 2000 (Y2K)

This test case will verify that MDTXT and MATXT correctly retrieves multiple textual observation messages identified in Table 3.2-7 of the METOC Database SRS. The test will demonstrate the retrieving of a multiple textual observation using a series of test cases. The textual observations which are part of the furnished ingest process have time stamps which cross from the year 1999 to 2000.

4.4.3.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

3. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

4. The database must have textual observation data available. It is required to ingest the simulated observations furnished with this program prior to running this test case.

4.4.3.2 Test Inputs

The test inputs necessary for this test case are provided in the GET test case driver files described in **Appendix A** of this document. These fields cover, Type, Report Times, Valid Times, Lat/Lon, Originating Site, and the Method of Receipt, which are used in combination to retrieve multiple textual observations from the database.

4.4.3.3 Expected Test Results

The API will retrieve correctly formatted textual observation messages and associated descriptive data corresponding to the input textual observation messages. The detailed expected test results are provided in **Appendix B** of this document.

4.4.3.4 Criteria for Evaluating Results

The test case outputs derived from the test driver must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.4.3.5 Test Procedure

Once the Prerequisite Conditions (4.4.3.1) for the test have been met, the **MATXTtestGetByQuery** test driver program is run by the tester. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.4.3.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.4.4 Retrieve Textual Observations with Get Test Case, Wild Card

This test case will verify that MDTXT and MATXT correctly retrieves multiple textual observation messages identified in Table 3.2-7 of the METOC Database SRS. The test will demonstrate the retrieving of a multiple textual observation using a series of test cases that query a set of inputs which are stamped as a wild card (*).

4.4.3.7 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows

are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document.

3. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

4. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

5. The database must have textual observation data available. It is required to ingest the simulated observations furnished with this program prior to running this test case.

4.4.3.8 Test Inputs

The test inputs necessary for this test case are provided in the **GET** test case driver files described in **Appendix A** of this document. These fields cover, Type, Report Times, Valid Times, Lat/Lon, Originating Site, and the Method of Receipt, which are used in combination to retrieve multiple textual observations from the database.

4.4.3.9 Expected Test Results

The API will retrieve correctly formatted textual observation messages and associated descriptive data corresponding to the input textual observation messages. The detailed expected test results are provided in **Appendix B** of this document.

4.4.3.10 Criteria for Evaluating Results

The test case outputs derived from the test driver must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.4.3.11 Test Procedure

Once the Prerequisite Conditions (4.4.4.1) for the test have been met, the **MATXTtestGet** test driver program is run by the tester. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.4.3.12 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.4.4 Simultaneous Retrieve Textual Observations with Get By ID Test Case

This test case will verify that MDTXT and MATXT correctly retrieves multiple textual observation messages identified in Table 3.2-7 of the METOC Database SRS when executed from the HP-UX and Windows NT machines simultaneously. The test will demonstrate the retrieving of a multiple textual observation on both systems without error using a series of identical test cases.

4.4.4.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document.

3. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

4. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

5. The database must have textual observation data available. It is required to ingest the simulated observations furnished with this program prior to running this test case.

4.4.4.2 Test Inputs

The test inputs necessary for this test case are provided in the **GET** test case driver files described in **Appendix A** of this document. These fields cover, Type, Report Times, Valid Times, Lat/Lon, Originating Site, and the Method of Receipt, which are used in combination to retrieve multiple textual observations from the database.

4.4.4.3 Expected Test Results

The API will retrieve correctly formatted textual observation messages and associated descriptive data corresponding to the input textual observation messages. The detailed expected test results are provided in **Appendix B** of this document.

4.4.4.4 Criteria for Evaluating Results

The test case outputs derived from the test driver must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.4.4.5 Test Procedure

Once the Prerequisite Conditions (4.4.5.1) for the test have been met, the **MATXTtestGetByID** test driver program is run by the tester on both the HP-UX and Windows NT machines simultaneously. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.4.4.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.5 Textual Observation Catalog Listing Test

The following test cases verify that the MDTXT database and MATXT API segments supporting the retrieval of multiple list of textual observations from the database.

4.5.1 Retrieve Textual Observations Catalog Listing with Catalog Test Case

This test case will verify that MDTXT and MATXT correctly retrieves a catalog listing of textual observation messages identified in Table 3.2-7 of the METOC Database SRS. The test will demonstrate the retrieving of a catalog list of textual observation using a series of test cases.

4.5.1.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document.

3. The database server must be "Up" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting Identify Storage under the Database Storage menu bar.

4. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "Up" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

5. The database must have textual observation data available. It may be desirable to run a simulated or real data ingest prior to running this test case.

4.5.1.2 Test Inputs

The test inputs necessary for this test case are provided in the CAT test case driver files described in **Appendix A** of this document. These files provide data required to query each input field. These fields cover, Type, Report Times, Valid Times, Lat/Lon, Originating Site, and the Method of Receipt, which are used in combination to retrieve a listing of textual observations from the database.

4.5.1.3 Expected Test Results

The API will retrieve a listing of textual observation messages corresponding to the input textual observation messages. The detailed expected test results are provided in **Appendix B** of this document.

4.5.1.4 Criteria for Evaluating Results

The test case outputs derived from the test driver must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.5.1.5 Test Procedure

Once the Prerequisite Conditions (4.5.1.1) for the test have been met, the **MATXTtestCatalog** test driver program is run by the tester. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.5.1.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.5.2 Retrieve Textual Observations with the Catalog, Geographic Area Test Case

This test case will verify that MDTXT and MATXT correctly retrieves a catalog listing of textual observation messages identified in Table 3.2-7 of the METOC Database SRS. The test will demonstrate the retrieving of a listing of available textual observation using a series of test cases. These test cases are geared to exercise the retrieval of specific geographic areas, which may be confusing for the database especially when making requests across the equator, international date, line and the Prime Meridian.

4.5.2.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document.

3. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

4. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

5. The database must have textual observation data available. It may be desirable to run a simulated or real data ingest prior to running this test case.

4.5.2.2 Test Inputs

The test inputs necessary for this test case are provided in the **CAT** test case driver files described in **Appendix A** of this document. These fields cover, Type, Report Times, Valid Times, Lat/Lon, Originating Site, and the Method of Receipt, which are used in combination to retrieve a catalog listing of textual observations from the database.

4.5.2.3 Expected Test Results

The API will retrieve a listing of textual observation messages corresponding to the input textual observation messages. The detailed expected test results are provided in **Appendix B** of this document.

4.5.2.4 Criteria for Evaluating Results

The test case outputs derived from the test driver must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.5.2.5 Test Procedure

Once the Prerequisite Conditions (4.5.2.1) for the test have been met, the **MATXTtestCatalog** test driver program is run by the tester. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.5.2.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.5.3 Retrieve Textual Observations with the Get Catalog Test Case, Year 2000 (Y2K)

This test case will verify that MDTXT and MATXT correctly retrieves a catalog listing of textual observation messages identified in Table 3.2-7 of the METOC Database SRS. The test will demonstrate the retrieving of a listing of available textual observation using a series of test cases. The textual observations which are part of the furnished ingest process have time stamps which cross from the year 1999 to 2000.

4.5.3.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document.

3. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

4. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

5. The database must have textual observation data available. It is required to ingest the simulated observations furnished with this program prior to running this test case.

4.5.3.2 Test Inputs

The test inputs necessary for this test case are provided in the **CAT** test case driver files described in **Appendix A** of this document. These fields cover, Type, Report Times, Valid Times, Lat/Lon, Originating Site, and the Method of Receipt, which are used in combination to retrieve a catalog listing of textual observations from the database.

4.5.3.3 Expected Test Results

The API will retrieve a listing of textual observation messages corresponding to the input textual observation messages. The detailed expected test results are provided in **Appendix B** of this document.

4.5.3.4 Criteria for Evaluating Results

The test case outputs derived from the test driver must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.5.3.5 Test Procedure

Once the Prerequisite Conditions (4.5.3.1) for the test have been met, the **MATXTtestCat** test driver program is run by the tester. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.5.3.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.5.4 Retrieve Textual Observations with the Catalog Test Case, Wild Card

This test case will verify that MDTXT and MATXT correctly retrieves a catalog listing of textual observation messages identified in Table 3.2-7 of the METOC Database SRS. The test will demonstrate the retrieving of a listing of available textual observation using a series of test cases that query a set of inputs which are stamped as a wild card (*).

4.5.4.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows

are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

3. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

4. The database must have textual observation data available. It is required to ingest the simulated observations furnished with this program prior to running this test case.

4.5.4.2 Test Inputs

The test inputs necessary for this test case are provided in the **CAT** test case driver files described in **Appendix A** of this document. These fields cover, Type, Report Times, Valid Times, Lat/Lon, Originating Site, and the Method of Receipt, which are used in combination to retrieve a catalog listing of textual observations from the database.

4.5.4.3 Expected Test Results

The API will retrieve a listing of textual observation messages corresponding to the input textual observation messages. The detailed expected test results are provided in **Appendix B** of this document.

4.5.4.4 Criteria for Evaluating Results

The test case outputs derived from the test driver must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.5.4.5 Test Procedure

Once the Prerequisite Conditions (4.5.4.1) for the test have been met, the **MATXTtestCatalog** test driver program is run by the tester. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.5.4.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.5.5 Retrieve Textual Observations with the Catalog Error Test Case

This test case will verify that MDTXT and MATXT will not retrieves a catalog listing of textual observation messages identified in Table 3.2-7 of the METOC Database SRS that has erroneous query data fields. The test will demonstrate that a variety of erroneous data fields will not retrieve the desired catalog listing of textual observations using a series of test cases.

4.5.5.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.
2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this documents. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.
3. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

4. The database must have textual observation data available. It is required to ingest the simulated observations furnished with this program prior to running this test case.

4.5.5.2 Test Inputs

The test inputs necessary for this test case are provided in the **CAT** test case driver files described in **Appendix A** of this document. These fields cover, Type, Report Times, Valid Times, Lat/Lon, Originating Site, and the Method of Receipt, which are used in combination to verify that erroneous data will not produce a catalog listing of textual observations from the database.

4.5.5.3 Expected Test Results

The API will not retrieve a listing of available observation messages and associated descriptive data corresponding to the input textual observation messages. The detailed expected test results are provided in **Appendix B** of this document.

4.5.5.4 Criteria for Evaluating Results

The test case outputs derived from the test driver must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.5.5.5 Test Procedure

Once the Prerequisite Conditions (4.5.5.1) for the test have been met, the **MATXTtestGet** test driver program is run by the tester. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.5.5.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.5.6 Simultaneous Retrieve Textual Observations with the Catalog Test Case

This test case will verify that MDTXT and MATXT correctly retrieves a catalog listing textual observation messages identified in Table 3.2-7 of the METOC Database SRS when executed from the HP-UX and the Windows NT machines simultaneously. The test will demonstrate the retrieving of a catalog listing of textual observation on both systems without error using a series of identical test cases.

4.5.6.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document.

3. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

4. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

5. The database must have textual observation data available. It is required to ingest the simulated observations furnished with this program prior to running this test case.

4.5.6.2 Test Inputs

The test inputs necessary for this test case are provided in the **CAT** test case driver files described in **Appendix A** of this document. These fields cover, Type, Report Times, Valid Times, Lat/Lon, Originating Site, and the Method of Receipt, which are used in combination to retrieve a catalog listing of textual observations from the database.

4.5.6.3 Expected Test Results

The API will retrieve a catalog listing of textual observation messages and associated descriptive data corresponding to the input textual observation messages. The detailed expected test results are provided in **Appendix B** of this document.

4.5.6.4 Criteria for Evaluating Results

The test case outputs derived from the test driver must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.5.6.5 Test Procedure

Once the Prerequisite Conditions (4.5.6.1) for the test have been met, the **MATXTtestCatalog** test driver program is run by the tester on both the HP-UX and Windows NT machines simultaneously. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.5.6.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.6 Updating an Existing Textual Observation Test

The following test cases verify that the MDTXT database and MATXT API segments supporting the update of an existing textual observation. This test will also verify multi-usage and updates using erroneous data.

4.6.1 Updating an Existing Textual Observation Test Case

This test case will verify that MDTXT and MATXT correctly updates textual observation data identified in Table 3.2-7 of the METOC Database SRS. The test will demonstrate the updating of the data fields of existing observations in the database. As a result, a new observation will be made and it will be marked as edited.

4.6.1.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document.

3. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

4. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

5. The database must have textual observation data available. It may be desirable to run a simulated or real data ingest prior to running this test case.

4.6.1.2 Test Inputs

The test inputs necessary for this test case are provided in the **UPDATE** test case driver files described in **Appendix A** of this document. These files provide data required to query each input field. These fields cover, Type, Report Times, Valid Times, Lat/Lon, Originating Site, and the Method of Receipt, which are used in combination to update the data within the textual observations of the database.

4.6.1.3 Expected Test Results

The API will update the textual observation messages corresponding to the input changes. The detailed expected test results are provided in **Appendix B** of this document.

4.6.1.4 Criteria for Evaluating Results

The test case outputs derived from the test driver and tester makes use of the database access tool (e.g., Informix DB Access discussed in Section 3.1.3) must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The database is directly test using the Date Base Access (DB Access) tool to query and verify observation data entries have been updated properly. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.6.1.5 Test Procedure

Once the Prerequisite Conditions (4.6.1.1) for the test have been met, the **MATXTtestUpdate** test driver program is run by the tester. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.6.1.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.6.2 Updating an Existing Textual Observation with Erroneous Data Test Case

This test case will verify that MDTXT and MATXT will not update textual observation data identified in Table 3.2-7 of the METOC Database SRS. The test will demonstrate that erroneous data will not update the data fields of existing observations in the database. As a result, the user will be informed that any data entry that does not fit the data structure will be unexceptable.

4.6.2.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document.

3. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

4. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

5. The database must have textual observation data available. It may be desirable to run a simulated or real data ingest prior to running this test case.

4.6.2.2 Test Inputs

The test inputs necessary for this test case are provided in the **UPDATE** test case driver files described in **Appendix A** of this document. These files provide data required to query each input field. These fields cover, Type, Report Times, Valid Times, Lat/Lon, Originating Site, and the Method of Receipt, which are used in combination to update the data within the textual observations of the database.

4.6.2.3 Expected Test Results

The API will update the textual observation messages corresponding to the input changes. The detailed expected test results are provided in **Appendix B** of this document.

4.6.2.4 Criteria for Evaluating Results

The test case outputs derived from the test driver and tester makes use of the database access tool (e.g., Informix DB Access discussed in Section 3.1.3) must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The database is directly test using the Date Base Access (DB Access) tool to query and verify observation data entries have been updated properly. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.6.2.5 Test Procedure

Once the Prerequisite Conditions (4.6.2.1) for the test have been met, the **MATXTtestUpdate** test driver program is run by the tester. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.6.2.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.6.3 Simultaneous Updating of Existing Textual Observations Test Case

This test case will verify that MDTXT and MATXT correctly update textual observation data identified in Table 3.2-7 of the METOC Database SRS when executed from both the HP-UX and the Windows NT machines simultaneously. The test will demonstrate that updating data fields of existing observations in the database on both systems without error using a series of identical test cases.

4.6.3.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document.

3. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

4. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

5. The database must have textual observation data available. It may be desirable to run a simulated or real data ingest prior to running this test case.

4.6.3.2 Test Inputs

The test inputs necessary for this test case are provided in the **Update** test case driver files described in **Appendix A** of this document. These files provide data required to query each input field. These fields cover, Type, Report Times, Valid Times, Lat/Lon, Originating Site, and the Method of Receipt, which are used in combination to update the data within the textual observations of the database.

4.6.3.3 Expected Test Results

The API will update the textual observation messages corresponding to the input changes. The detailed expected test results are provided in **Appendix B** of this document.

4.6.3.4 Criteria for Evaluating Results

The test case outputs derived from the test driver and tester makes use of the database access tool (e.g., Informix DB Access discussed in Section 3.1.3) must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The database is directly test using the Date Base Access (DB Access) tool to query and verify observation data entries have been updated properly. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.6.3.5 Test Procedure

Once the Prerequisite Conditions (4.6.3.1) for the test have been met, the **MATXTtestUpdate** test driver program is run by the tester on both the HP-UX and Windows NT machines simultaneously. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.6.3.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.7 Deleting a Textual Observation Test

The following test cases verify that the MDTXT database and MATXT API segments supporting the deletion of an existing textual observation. This test will also verify multi-usage and deletions using erroneous data.

4.7.1 Deleting a Textual Observation Test Case

This test case will verify that MDTXT and MATXT correctly deletes a single textual observation messages identified in Table 3.2-7 of the METOC Database SRS. The test will demonstrate the deletion of a single observation using a series of test cases.

4.7.1.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document.

3. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

4. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

5. The database must have textual observation data available. It may be desirable to run a simulated or real data ingest prior to running this test case.

4.7.1.2 Test Inputs

The test inputs necessary for this test case are provided in the DELETE test case driver files described in **Appendix A** of this document. These files provide the Record ID required for deletion of a single textual observations from the database.

4.7.1.3 Expected Test Results

The API will delete a textual observation messages (one per case) and associated descriptive data corresponding to the input textual observation messages. The detailed expected test results are provided in **Appendix B** of this document.

4.7.1.4 Criteria for Evaluating Results

The test case outputs derived from the test driver and tester use of the database access tool (e.g., Informix DB Access discussed in Section 3.1.3) must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The database is directly test using the Date Base Access (DB Access) tool to query and verify observation data entries have been updated properly. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.7.1.5 Test Procedure

Once the Prerequisite Conditions (4.7.1.1) for the test have been met, the **MATXTtestDeleteByID** test driver program is run by the tester. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.7.1.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.7.2 Deleting a Textual Observation with the Delete By ID Error Test Case

This test case will verify that MDTXT and MATXT will not delete a single textual observation messages identified in Table 3.2-7 of the METOC Database SRS that has an erroneous Record ID. The test will demonstrate that an erroneous Record ID will not delete a single observation using a series of test cases.

4.7.2.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

3. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

4. The database must have textual observation data available. It may be desirable to run a simulated or real data ingest prior to running this test case.

4.7.2.2 Test Inputs

The test inputs necessary for this test case are provided in the **DELETE** test case driver files described in **Appendix A** of this document. These files provide the Record ID required for deleting a single textual observations from the database.

4.7.2.3 Expected Test Results

The API will not delete a textual observation messages (one per case) and associated descriptive data corresponding to the input textual observation messages. The detailed expected test results are provided in **Appendix B** of this document.

4.7.2.4 Criteria for Evaluating Results

The test case outputs derived from the test driver and tester makes use of the database access tool (e.g., Informix DB Access discussed in Section 3.1.3) must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The database is directly test using the Date Base Access (DB Access) tool to query and verify observation data entries have been updated properly. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.7.2.5 Test Procedure

Once the Prerequisite Conditions (4.7.2.1) for the test have been met, the **MATXTtestDeleteByID** test driver program is run by the tester. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.7.2.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

4.7.3 Simultaneous Deleting a Textual Observation with the Delete By ID Test Case

This test case will verify that MDTXT and MATXT correctly deletes a single textual observation messages identified in Table 3.2-7 of the METOC Database SRS when executed from the HP-UX and Windows NT machine simultaneously. The test will demonstrate the deletion of a single observation on both systems with out error using a series of identical test cases. For this particular case either the HP-UX or Windows NT will attempt to delete the requested observation,

but only one will be successful. The other platform will receive an error that the observation does not exist.

4.7.3.1 Prerequisite Conditions

The following conditions must be established prior to executing this test case:

1. Applicable segments are loaded on the target test machines as specified in Section 3.0 of this document.

2. Testing should be conducted while logged in as *sysadmin*. On the HP target platform the tester must be able to initiate and have access to an xterm window. On the Windows NT machine, the tester must be able to initiate and have access to a DOS window. These windows are required to operate the test driver programs and capture output. Procedures for establishing these windows are provided in Section 3.1.3 of this document. The database server must be "**Up**" for proper connectivity between the API and Database. This can be verified by logging in as *dbadmin* and selecting **Identify Storage** under the Database Storage menu bar.

3. To ensure the test environment has a clean database, it is recommended that database and API segments be deinstalled and the reinstalled. This will clean out any data on the database server. Note: The database server should be in an "**Up**" status prior to install/deinstalls. Refer to the segment installation procedures (Section 3.0 of this document) for proper installation procedures.

4. The database must have textual observation data available. It may be desirable to run a simulated or real data ingest prior to running this test case.

4.7.3.2 Test Inputs

The test inputs necessary for this test case are provided in the **DELETE** test case driver files described in **Appendix A** of this document. These files provide the Record ID required for retrieval of a single textual observations from the database.

4.7.3.3 Expected Test Results

The API will delete a formatted textual observation messages (one per case) and associated descriptive data corresponding to the input textual observation messages. The detailed expected test results are provided in **Appendix B** of this document.

4.7.3.4 Criteria for Evaluating Results

The test case outputs derived from the test driver and tester makes use of the database access tool (e.g., Informix DB Access discussed in Section 3.1.3) must match exactly with the Expected Test Results. The range of accuracy is exact with no deviation from the expected result. The output information is compared by re-directing the results to a file using the test driver debug command line argument or conventional screen capture techniques. The file is then viewed and compared with the expected test results provided in this document. The database is directly test using the Date Base Access (DB Access) tool to query and verify observation data entries have been updated properly. The test case must be run on both target platforms (HP-UX and Windows NT) and the test results compared to ensure both platforms provide the same results.

4.7.3.5 Test Procedure

Once the Prerequisite Conditions (4.7.3.1) for the test have been met, the **MATXTtestDeleteByID** test driver program is run by the tester on both the HP-UX and Windows NT machines simultaneously. The test driver program automatically executes the specified test case. Section 3.1.2 of this document describes the procedures required to run the test driver program and obtain the test results.

4.7.3.6 Assumptions and Constraints

This test case assumes the target hardware is operating correctly and configured with the operating, application, and test driver software identified in Sections 3.1 and 3.2 of the MDTXT IP and this document.

5 REQUIREMENTS TRACEABILITY

All of the test cases discussed in Section 4 of this document were derived from the specifications and requirements referenced in the *Performance Specification (PS) for the Tactical Environmental Support System/Next Century [TESS(NC)] (UN/UMK-3)* and the *Software Requirements Specification for the Tactical Environmental Support System/Next Century [TESS(NC)] Meteorological and Oceanographic (METOC) Database*.

6 NOTES

6.1 Glossary of Acronyms

AESS	Allied Environmental Support System
API	Application Program Interface
APIRM	API Reference Manual
COE	Common Operating Environment
DBDD	Database Design Description
DID	Data Item Description
DII	Defense Information Infrastructure
GCCS	Global Command and Control System
IC4ISR	Integrated Command, Control, Communications, Computer, and Intelligence Surveillance Reconnaissance
IMOSS	Interim Mobile Oceanographic Support System
JMCIS	Joint Maritime Command Information System
JMS	Joint METOC Segment
MATXT	Textual Observation API Segment of the TESS(NC) METOC Database
MDTXT	Textual Observation Database Segment of the TESS(NC) METOC Database
METOC	Meteorological and Oceanographic
MIDDS	Meteorological Integrated Data Display System
NC	Next Century
PS	Performance Specification
SRS	Software Requirements Specification

TESS	Tactical Environmental Support System

Appendix A - Textual Observation Segment Test Inputs

The following data and files are required as input data by textual observation segment (i.e. MDTXT, MATXT) testing.

A.1 MDTXT and MATXT Segment Installation Test

No input data is required by these Test Cases 4.1.1, 4.1.2, and 4.1.3. See associated MDTXT and MATXT Installation Procedure documents.

A.2 Textual Observation Data Ingest Test

A.2.1 Ingest Required Textual Observations Test Case (4.2.1) Inputs

The following files provide the test case data required as inputs. These files and the contents of these files can be found in the **/h/MATXT/Integ/TestSuite/ TESTDATA/INGEST** directory on the target platform.

Input Filename	Description
for_ext	Forecast Report Type - Extended
for_guide	Forecast Report Type - Guidance
for_icebg_NPole	Forecast Report Type - Iceberg
for_local_y2k	Forecast Report Type - Local with Y2K check
for_misc	Forecast Report Type - Miscellaneous
for_othship_world	Forecast Report Type - Other Shipping
for_rms	Forecast Report Type- Radio Warning Service
for_tmpExt	Forecast Report Type- Temperature Extremes
for_tropCycAdv	Forecast Report Type- Tropical Cyclone Advisories
for _ua_equator	Forecast Report Type- Upper Air Thickness with geographic position check over the equator
for_volAsh	Forecast Report Type- Volcanic Ash
for_wtrSprts	Forecast Report Type- Winter Sports
not_METNOWIFMA_world	Notices Type- METNO/WIFMA
not_hydro_equator	Notices Type- Hydrological with geographic position check over the equator
not_marine	Notices Type- Marine
not_nuclear	Notices Type- Nuclear Emergency
not_prodGenDelay	Notices Type- Product Generation Delay

Input Filename	Description
not_testMsg_world	Notices Type- Test Message with geographic position check of the entire world
not_warnOrCancel	Notices Type- Warning Related or Cancellation
srf_misc_equator	Surface Report Type- Miscellaneous with geographic position check of the entire world
srf_seaIce_SPole	Surface Report Type- Sea Ice with geographic position check of the South Pole
srf_snoDep	Surface Report Type- Snow Depth
ua_misc	Upper Air Type- Miscellaneous
warn_SIGMET	Surface Report Type- SIGMET
warn_airSig	Surface Report Type- AIRMET/SIGMET
Warn_hiSeasUSAF_dateLine	Surface Report Type- High Seas (USAF) with geographic position check over the International Date Line
warn_milUSAF	Surface Report Type- Military Weather Warnings (USAF)
warn_miscUSAF	Surface Report Type- Misc. Weather Warnings (USAF)
warn_other_y2k	Surface Report Type- Other with Y2K check
warn_rivFld_world	Surface Report Type- River Flood with geographic position check of the entire world
warn_sevThundr	Surface Report Type- Severe Thunderstorm
warn_sum	Surface Report Type- Warnings and Summaries
warn_tornUSAF	Surface Report Type- Tornado (USAF)
warn_tropCycDis	Surface Report Type- Tropical Cyclone Discussion
warn_tropCycSig	Surface Report Type- Tropical Cyclone (SIGMET)
warn_tropCycTyph_point	Surface Report Type- Tropical Cyclone (Typhoon) with geographic position check of single point
warn_tsnuami	Surface Report Type- Tsunami
warn_volAshSIG	Surface Report Type- Volcanic Ash (SIGMET)
x_bad_area	Report with a unrecognizable geographic area to check the handling of erroneous data
x_bad_time	Report with a unrecognizable time stamps to check the handling of erroneous data
x_bad_type	Report with a unrecognizable Textual Observation type to check the handling of erroneous data

The following is an example of the data structure required for the ingesting of a typical textual observation:

F	TextObsType (See Table 3.2-7 of the SRS)
E	TextObsSubType (See Table 3.2-7 of the SRS)
0	QualityIndicator
0302199800	ReportTime (MMDDYYYYHH)
0302199806	BegValidTime (MMDDYYYYHH)
0302199812	EndValidTime (MMDDYYYYHH)
90.000000	North Latitude
45.000000	South Latitude
-180.000000	West Longitude
90.000000	East Longitude
1	DataCategory
UNCLASS	SecurityClass (text)
KUUU	OriginatingSite (text)
Hard Copy	ReceiptMethod (text)
String	Ingest_string (file name with text)

A.2.2 Multi-user Textual Observations Ingest Test Case (4.2.2) Inputs

The multi-user test ingests all of the textual observations described in A.2.1. The only difference if that the ingest test driver is initiated simultaneously at the HP and Windows NT machines. The database is then checked to verify that duplicate textual observations have been ingested properly.

A.3 Textual Observation Get By ID Test

A.3.1 Retrieve a Textual Observation with the Get By ID Test Case (4.3.1) Inputs

The following files provide the test case data required as inputs. These files and the contents of these files can be found in the **/h/MATXT/Integ/TestSuite/ TESTDATA/GETBYID** directory on the target platform.

Input Filename	Description
getId21	Retrieves an observation with a Record ID of 21
getId23	Retrieves an observation with a Record ID of 23
getId4	Retrieves an observation with a Record ID of 4
getId5	Retrieves an observation with a Record ID of 5
getId9	Retrieves an observation with a Record ID of 9

The following is an example of the data structure required for retrieving a typical textual observation using the Get By ID:

21 Record ID

A.3.2 Retrieve a Textual Observation with the Get By ID Error Test Case (4.3.1) Inputs

The following files provide the test case data required as inputs. These files and the contents of these files can be found in the **/h/MATXT/Integ/TestSuite/ TESTDATA/GETBYID** directory on the target platform.

Input Filename	Description
getId-6	Attempts to retrieves an observation with an erroneous Record ID of -6
getId200	Attempts to retrieves an observation with an erroneous Record ID of 200 (This should not work if less than 200 textual observations are in the database)

A.3.3 Simultaneous Retrieve of a Textual Observations with the Get By ID Test Case (4.3.3) Inputs

The multi-user test retrieves all of the textual observations described in A.3.1. The only difference if that the Get By ID test driver is initiated simultaneously at the HP and Windows NT machines.

A.4 Textual Observation Get By Query Test

A.4.1 Retrieve Textual Observations with the Get By Query Test Case Inputs

The following files provide the test case data required as inputs. These files and the contents of these files can be found in the **/h/MATXT/Integ/TestSuite/ TESTDATA/GET** directory on the target platform.

Input Filename	Description
all_forecasts	Retrieves all Forecast Reports with wild card entries
all_warnings	Retrieves all Warnings with wild card entries
all_notices	Retrieves all Notices with wild card entries
all_surface	Retrieves all Surface Reports with wild card entries
all_upper_air	Retrieves all Upper Air Obs with wild card entries
upper_air_match	Retrieves Upper Air Obs that are an exact match to the query criteria
for_ext_match	Retrieves Extended Forecast Reports that are an exact match to the query criteria

The following is an example of the data structure required for retrieving a one or more textual observations.

F	TextObsType (See Table 3.2-7 of the SRS)
E	TextObsSubType (See Table 3.2-7 of the SRS)
0302199800	BegReportTime (MMDDYYYYHH)
0302199800	EndReportTime (MMDDYYYYHH)
0302199806	BegValidTime (MMDDYYYYHH)
0302199812	EndValidTime (MMDDYYYYHH)
90.000000	North Latitude
45.000000	South Latitude
-180.000000	West Longitude
90.000000	East Longitude
KUUU	OriginatingSite (text)
Electronic	ReceiptMethod (text)

A.4.2 Retrieve Textual Observations with the Get By Query Test Case, Geographic Area (4.4.2) Inputs

The following files provide the test case data required as inputs. These files and the contents of these files can be found in the **/h/MATXT/Integ/TestSuite/ TESTDATA/GET** directory on the target platform.

Input Filename	Description
not_equator	Retrieves Notices covering the Equator
warn_point	Retrieves Warnings covering a geographic single point
for_NPole	Retrieves Forecast Reports covering the North Pole
srf_SPole	Retrieves Surface Reports covering the South Pole
warn_dateLine	Retrieves Warnings covering the International Date Line

A.4.3 Retrieve Textual Observations with the Get By Query Test Case, Y2K (4.4.3) Inputs

The following files provide the test case data required as inputs. These files and the contents of these files can be found in the **/h/MATXT/Integ/TestSuite/ TESTDATA/GET** directory on the target platform.

Input Filename	Description
warn_y2k	Retrieves Warnings filling the Y2K criteria
for_y2k	Retrieves Forecast Report filling the Y2K criteria

A.4.4 Retrieve Textual Observations with the Get By Query Test Case, Wild Card (4.4.4) Inputs

The following files provide the test case data required as inputs. These files and the contents of these files can be found in the **/h/MATXT/Integ/TestSuite/ TESTDATA/GET** directory on the target platform. Note, All of these files were used as part of a blanket retrieve discussed in (A.4.1).

Input Filename	Description
all_forecasts	Retrieves all Forecast Reports with wild card entries
all_warnings	Retrieves all Warnings with wild card entries
all_notices	Retrieves all Notices with wild card entries
all_surface	Retrieves all Surface Reports with wild card entries
all_upper_air	Retrieves all Upper Air Obs with wild card entries

The following is an example of the data structure required for retrieving a one or more textual observations. The "*" show those areas which maybe wild carded.

N	TextObsType (See Table 3.2-7 of the SRS)
*	TextObsSubType (See Table 3.2-7 of the SRS)
*	BegReportTime (MMDDYYYYHH)
*	EndReportTime (MMDDYYYYHH)
*	BegValidTime (MMDDYYYYHH)
*	EndValidTime (MMDDYYYYHH)
90.0	North Latitude
-90.0	South Latitude
-180.0	West Longitude
180.0	East Longitude
*	OriginatingSite (text)
*	ReceiptMethod (text)

A.4.5 Simultaneous Retrieve of a Textual Observations with the Get By ID Test Case (4.3.5) Inputs

The multi-user test retrieves all of the textual observations described in A.4.1. The only difference if that the Get By ID test driver is initiated simultaneously at the HP and Windows NT machines.

A.5 Textual Observation Catalog Listing Test

A.5.1 Retrieve Textual Observations Catalog Listing with the Catalog Test Case

The following files provide the test case data required as inputs. These files and the contents of these files can be found in the **/h/MATXT/Integ/TestSuite/ TESTDATA/CAT** directory on the target platform.

Input Filename	Description
all_forecasts	Retrieves a list of all Forecast Reports with wild card entries
receipt_ftp	Retrieves a list of Obs with an "ftp" as the method of receipt
site_KUUU	Retrieves a list of Obs with "KUUU" as the designate originating site
upper_air	Retrieves a list of Upper Air Obs that fits the specified criteria

The following is an example of the data structure required for retrieving a one or more textual observations.

F	TextObsType (See Table 3.2-7 of the SRS)
E	TextObsSubType (See Table 3.2-7 of the SRS)
0302199800	BegReportTime (MMDDYYYYHH)
0302199800	EndReportTime (MMDDYYYYHH)
0302199806	BegValidTime (MMDDYYYYHH)
0302199812	EndValidTime (MMDDYYYYHH)
90.000000	North Latitude
45.000000	South Latitude
-180.000000	West Longitude
90.000000	East Longitude
KUUU	OriginatingSite (text)
Electronic	ReceiptMethod (text)

A.5.2 Retrieve Textual Observations with the Get By Query Test Case, Geographic Area (4.5.2) Inputs

The following files provide the test case data required as inputs. These files and the contents of these files can be found in the **/h/MATXT/Integ/TestSuite/ TESTDATA/GET** directory on the target platform.

Input Filename	Description
srf_equator	Retrieves a list of Surface Reports covering the Equator
warn_world	Retrieves a list of Warnings covering the entire world
lat_lon	Retrieves a list of Forecast Reports covering the North Pole
world_ftp	Retrieves a list of Obs covering the entire world with "ftp" as the method of receipt

A.5.3 Retrieve Textual Observations with the Catalog Test Case, Wild Card (4.5.3) Inputs

The following files provide the test case data required as inputs. These files and the contents of these files can be found in the **/h/MATXT/Integ/TestSuite/ TESTDATA/GET** directory on the target platform. Note: All of these files were used as part of a blanket retrieve of a catalog list discussed in A.5.1 and A.5.2.

Input Filename	Description
srf_equator	Retrieves a list of Surface Reports covering the Equator
warn_world	Retrieves a list of Warnings covering the entire world
lat_lon	Retrieves a list of Forecast Reports covering the North Pole
world_ftp	Retrieves a list of Obs covering the entire world with "ftp" as the method of receipt
site_KUUU	Retrieves a list of Obs with "KUUU" as the designate originating site

The following is an example of the data structure required for retrieving a one or more textual observations. The "*" show those areas which maybe wild carded.

N	TextObsType (See Table 3.2-7 of the SRS)
*	TextObsSubType (See Table 3.2-7 of the SRS)
*	BegReportTime (MMDDYYYYHH)
*	EndReportTime (MMDDYYYYHH)
*	BegValidTime (MMDDYYYYHH)
*	EndValidTime (MMDDYYYYHH)
90.0	North Latitude
-90.0	South Latitude
-180.0	West Longitude
180.0	East Longitude
*	OriginatingSite (text)
*	ReceiptMethod (text)

A.5.4 Retrieve Textual Observations with the Catalog Error Test Case (4.5.4) Inputs

The following files provide the test case data required as inputs. These files and the contents of these files can be found in the **/h/MATXT/Integ/TestSuite/ TESTDATA/GET** directory on the target platform.

Input Filename	Description
x_bad_time	Attempts to retrieve a list of observations with erroneous time stamps
x_lat_lon	Attempts to retrieve a list of observations with erroneous geographic coordinates

A.5.5 Simultaneous Retrieve of a Textual Observations with the Catalog Test Case (4.5.5) Inputs

The multi-user test retrieves all of the textual observations described in A.5.1. The only difference is that the Catalog test driver is initiated simultaneously at the HP and Windows NT machines.

A.6 Updating an Existing Textual Observation Data Test

A.6.1 Updating an Existing Observations Test Case (4.6.1) Inputs

The following files provide the test case data required as inputs. These files and the contents of these files can be found in the **/h/MATXT/Integ/TestSuite/ TESTDATA/UPDATE** directory on the target platform. The database is then checked to verify that the textual observation has been updated.

Input Filename	Description
qual_end_NSWE_15	Updates an observation with a new Quality Indicator, End Time, Geographic Coordinates, and Record ID
update_all	Updates all data fields of a specific observation
sub_report_S_1	Updates and observation with a new Sub-type, Report, Time, South Latitude and Record ID

A.6.2 Updating an Existing Textual Observation with Erroneous Data Test Case (4.6.2) Inputs

The following files provide the test case data required as inputs. These files and the contents of these files can be found in the **/h/MATXT/Integ/TestSuite/ TESTDATA/UPDATE** directory on the target platform.

Input Filename	Description
x_bad_time	Attempts to update a specific observation with an erroneous time stamp
x_bad_recno	Attempts to update a specific observation with an erroneous Record ID (negative number)

A.6.3 Simultaneous Updating of an Existing Textual Observation Test Case (4.6.3) Inputs

The multi-user test updates all of the textual observations described in A.6.1. The only difference if that the update test driver is initiated simultaneously at the HP and Windows NT machines. The database is then checked to verify that the duplicate textual observations have been updated properly.

A.7 Deleting a Textual Observation Data Test

A.7.1 Textual Observation Delete By ID Test Case (4.7.1) Inputs

The following files provide the test case data required as inputs. These files and the contents of these files can be found in the **/h/MATXT/Integ/TestSuite/ TESTDATA/DELETE** directory on the target platform. The database is then checked to verify that the textual observation has been deleted.

Input Filename	Description
delete1	Deletes a Textual Observation with a Record ID of 1
delete4	Deletes a Textual Observation with a Record ID of 4
delete5	Deletes a Textual Observation with a Record ID of 5
delete7	Deletes a Textual Observation with a Record ID of 7
delete8	Deletes a Textual Observation with a Record ID of 8
delete17	Deletes a Textual Observation with a Record ID of 17

The following is an example of the data structure required for retrieving a typical textual observation using the Get By ID:

21 Record ID

A.7.2 Textual Observation Delete By ID Test Case with Errors (4.7.2) Inputs

The following files provide the test case data required as inputs. These files and the contents of these files can be found in the **/h/MATXT/Integ/TestSuite/ TESTDATA/DELETE** directory on the target platform.

Input Filename	Description
delete-67	Attempts to delete a specific observation with an erroneous Record ID of -67
delete101	Attempts to delete a specific observation with an erroneous Record ID of 101 (This should not work if less than 101 textual observations are in the database)

A.7.3 Simultaneous Deleting a Textual Observation with the Delete By ID Test Case (4.7.3) Inputs

The multi-user test deletes all of the textual observations described in A.7.1. The only difference is that the delete test driver is initiated simultaneously at the HP and Windows NT machines. The database is then checked to verify that the textual observations have been removed.

Appendix B - Textual Observation Segment Expects and Report of Test Results

B.1 Test Results

For the MATXT and MDTXT segments, with the exception of the installation procedures, all testing was conducted with the constructed test cases described in **Appendix A**. It should be clarified that the test cases were developed in conjunction with the test drivers to assist the developers with a variety of fairly realistic data inputs and outputs. As a result of this, the test cases described in this document were continuously updated to ensure the end results matched the expected results. In doing so the test team worked closely with the development team to verify reasons for test cases that did not match the expected results.

For this reason the expected results are identical to the planned results when running the final predelivery tests. Test drivers and cases were verified on Configuration Managed (CM) HP-UX and Windows NT 4.0 platforms. Discrepancies were documented using a PTR database. Corrected PTRs were again tested in the CM environment to verify that problems or enhancements were properly resolved (Open and closed PTRs are listed in the Software Version Description documents for the MATXT and MDTXT segments).

B.2 Problems Encountered

The only major problem encountered with this series of test dealt with the test results output for the NT and HP-UX platforms. When running batch processing, the NT and HP systems will execute the test case files in a different sequence. As a result, when reviewing and comparing the output data for each test, the NT system will show the same results as the HP platform but in a different order or sequence.

B.3 Test Case Results

Due to large amount of output data and results when running the provided test cases in a batch processing mode, the data is provided on a 3.5" floppy disc in a "text" format. Test cases which meet this criteria are annotated in this section.

B.3.1 MDTXT and MATXT Segment Installation Test

Figure B-1 is a graphic example of the Segment Installer on the HP-UX system. Once the MD/MATXT segments are properly installed the user will see the segments listed in the Currently Installed Segments portion of the Installer window. In addition, the MD/MATXT segments will still be listed in the Select Software To Install portion of the Installer window. In this window the installed segments will have a "*" prior to each name. This denotes that the segment is successfully installed on the HP-UX platform.

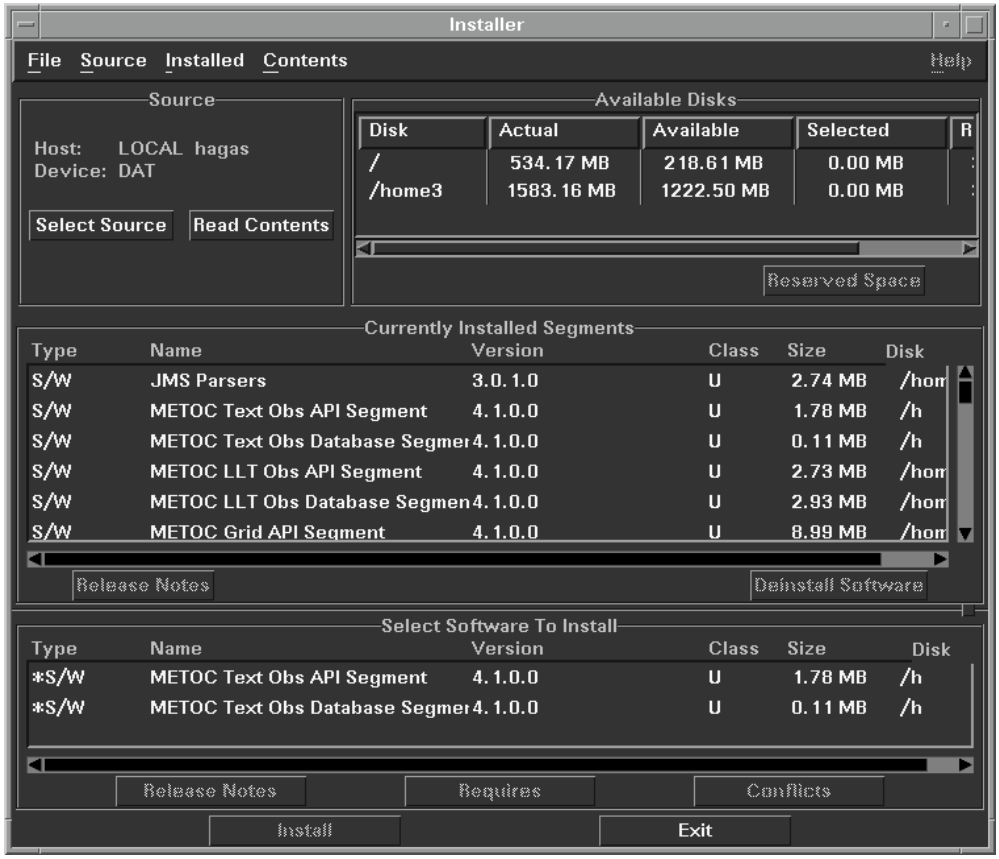

Figure B-1. HP-UX Segment Installer

Figure B-2 is a graphic example of the Install Shield on the Windows 4.0 NT system. Once installed, the Add/Remove Programs Properties window will have the MATXT segment listed in the list provided.

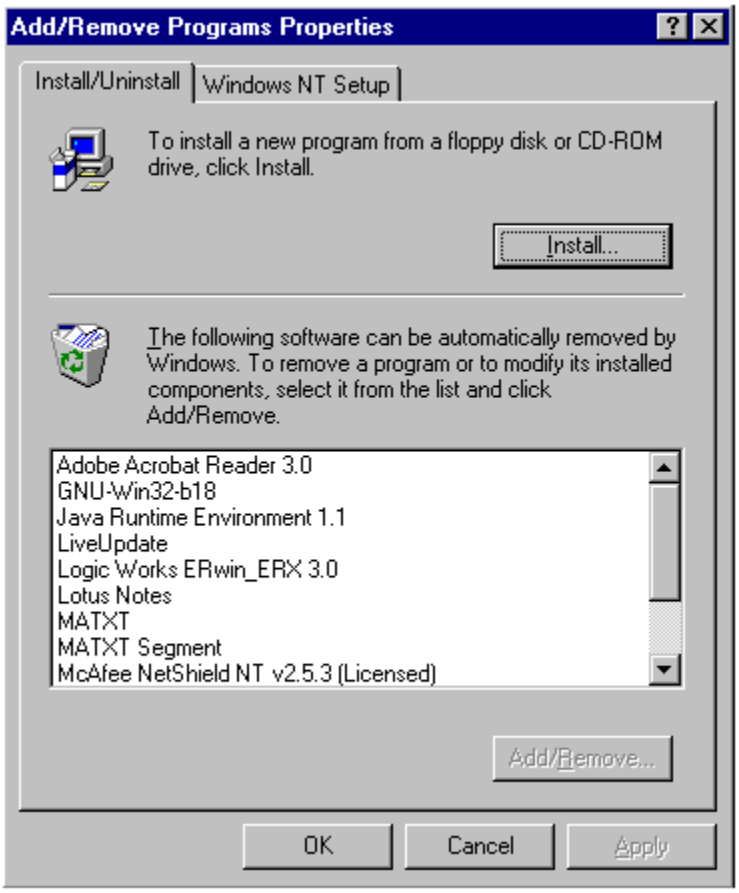

Figure B-1. Add/Remove Programs Properties Window

B.3.2 Textual Observation Data Ingest Test

The test data output and results are provided on the 3.5” floppy provided with this document. The file name is *Ingest*.

B.3.3 Textual Observation Get By ID Test

The test data output and results are provided on the 3.5” floppy provided with this document. The file name is *GetById*.

B.3.4 Textual Observation Get By Query Test

The test data output and results are provided on the 3.5" floppy provided with this document. The file name is *GetByQry*.

B.4.5 Textual Observation Catalog Listing Test

The test data output and results are provided on the 3.5" floppy provided with this document. The file name is *Catalog*.

B.4.6 Updating an Existing Textual Observation Data Ingest Test

The test data output and results are provided on the 3.5" floppy provided with this document. The file name is *Update*.

B.4.7 Deleting a Textual Observation Data Ingest Test

The test data output and results are provided on the 3.5" floppy provided with this document. The file name is *Delete*.